PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(54) Title: A METHOD OF AND APPARTUS FOR GENERATING DATA FOR CHARGING A USER FOR ACCESS OVER A COMMUNICATIONS NETWORK LINK

(57) Abstract

A method of generating charging data relating to information passed between a user's computer system and a content provider's computer system over a communications network link involving at least one client/server relationship, the method comprising the following steps: a) monitoring communications in at least one client/server relationship to detect at least one event causing a change in a state of at least one logical connection defined by at least a client network layer address, a client transport layer address, a server network layer address and a server transport layer address, b) recording data created in response to the at least one event detected by the monitoring, and c) generating charging data on the basis of the recorded data.

A METHOD OF AND APPARATUS FOR GENERATING DATA FOR CHARGING A USER FOR ACCESS OVER A COMMUNICATIONS NETWORK LINK

5

Field of the Invention

This invention relates to a method of and apparatus for generating data for charging a user of a computer system for access over a communications network link. More particularly, but not exclusively, this invention relates to a method of

10 and apparatus for generating data for charging a user of a computer system for access to the content provided by a content provider whether directly or via an access provider.

Background of the Invention

In the field of the Internet, both the access provider and content provider

15 are sometimes referred to as Internet Service Providers. In this document the term "content provider" refers to an operator of a computer system from which a user of a computer system might want to receive content. The term "access provider" refers to an operator of a computer system which, although it may not contain content useful to the user, enables the user to receive the content from a content

20 provider.

There are a number of technologies with which a user's computer system may be connected to the computer system of an access provider or a content provider over a communications link. One common technology is the "dial-up" modem. The two computer systems at each end of the link are attached to a

25 modem which serves to modulate or demodulate data sent or received so that it may be transmitted over an ordinary Public Switched Telephone Network (PSTN)

voice-grade telephone line. This technology, therefore, requires the user to dial up a telephone connection to the access provider or content provider. Another example of a "dial-up" technology is Integrated Services Digital Network (ISDN). This technology uses two ISDN terminal adapters at each end to split an ordinary

5   copper twisted-pair telephone line into two digital channels. It allows greater data transfer rates than ordinary voice-grade modem links, but still requires the user to dial-up a PSTN telephone connection.

Other technologies for physical communications links include permanently-activated communications links, otherwise known as "always-on" communications

10 links. One example of an "always-on" link is a leased line set up by a telecommunications company. Typically, leased lines are set up between geographically distant computer systems for a fixed monthly charge during which the link is, at least potentially, permanently-activated. Another example of communications link technologies which may be set up as "always-on" links are

15 digital subscriber lines, collectively referred to as xDSLs. xDSL communications links offer greater data transfer rates over copper twisted-pair telephone lines than, for example, ISDN communications links. Asymmetric Digital Subscriber Line (ADSL), one of the xDSL family, provides asymmetric data transfer capacities between the two terminal units connected over copper twisted-pair telephone lines

20 or other lines, so that the downstream data transfer rate from the content or access provider may be relatively high compared to the upstream data transfer rate from the user.

There are a number of common methods for charging the user of a computer system connected over a communications network link, for example to

25 the Internet, for the use he makes of the connection. Where the user's computer

system is connected to a content or access provider's computer system via a "dial-up" communications link, one method is to charge the user a monthly or yearly subscription fee which entitles him to a certain number of hours "on-line". An other method is to charge the user a monthly or yearly flat-rate fee for

5   unlimited "on-line" time. In both these common methods, the user normally also pays a telephone company for the duration of the telephone call to the content or access provider.

Where the user's computer system is connected to a content or access provider's computer system via an "always-on" communications link there is no

10   longer a meaningful measurement of the period of time the user spends "on-line". One method of charging the user is to charge a flat rate fee, optionally supplemented by a charge based on the throughput of data over the communications link.

One disadvantage of charging a flat-rate fee or a monthly or yearly

15   subscription fee for a certain number of hours "on-line" is that neither accurately charges for the amount of use made of the connection. In particular, these common solutions can cost disproportionate amounts for higher and lower usage users.

Measuring the throughput of data packets over a communications link

20   takes account of usage, but has a number of disadvantages as well. One disadvantage is that such measuring is a complex and processor-intensive task for the content or access provider since data packets can be numerous and of variable size.

In the prior art, International Patent Publication No. WO97 / 01920

25   describes a method for charging a user of a computer system for chargeable

services used, in which the user makes a separate telephone call to a predetermined chargeable telephone number which enables the connection between the user's computer system and the access or content provider. The charging is carried out on the basis of the telephone connection.

5   US patent no. 5,745,556 describes a payment mechanism for a user to access the Internet. A telephone company supplies an access or content provider with a telephone number in which charges may be set by the called party. Charging is carried out through an actual or virtual telephone call made to this telephone number and charged to the user.

10   US patent no. 5,737,414 describes a method of using an access management computer to control the user's access to data. The user connects to the access management computer over a "dial-up" communications link and obtains a unique access message either from the "homepage" of the access provider or from a separate telephone call made to the access provider. The

15 access provider also communicates to the user the charge for access to data based on the length of "on-line" connection time or some fixed charge.

International Patent Publication No. WO97 / 41586 describes a system capable of granting Internet access to users when direct connection to the usual access provider is impossible, impractical or prohibitively expensive. The user

20 connects to an independent access provider, over a "dial-up" communications link, which provides access to the Internet by validating the user's entitlement via a remote access control. Charging is effected by sending billing information packets at regular intervals to a co-ordinator.

European Patent Application No. EP 0-866,596-82 describes method and

25 apparatus for charging for Internet Telephony. When a caller connects to an

Internet access provider in order to make a voice call with another caller, the access provider sends information to a billing server concerning the identities of the callers, the beginning time of the call, the identity of the access provider and various set-up parameters of the call. During the call the billing server receives periodic updated information, and the access provider notifies the billing server at the end of the call. The billing server charges the appropriate caller based on the information provided to it which may include information on call routing, packet density and voice enhancement.

United States Patent No. 5,790,548 describes a system and method for providing access to services such as video on demand over a PSTN copper twisted-pair line using an "always-on" connection, such as ADSL. The document refers to methods of charging a user such as charging a flat fee or tracking the amount of user traffic throughput carried by the network.

In US patent 5 862 335, issued on 19th January 1999 (after the priority date of the present application), there is disclosed a method of and apparatus for monitoring logical connections and file transfers in a computer network. As a result of this monitoring, a database is generated recording information such as the identity of the end-point computers of a logical connection or file transfer, when connections or file transfers begin and end, and how much information is exchanged. This information may be used for various functions including billing computer users.

The monitoring is performed by a network monitor computer which is connected via an interface unit to the network and which intercepts all data packets exchanged by the network and creates a copy of them for analysis. A network management engine, running on the network monitoring computer,

analyses the copied data packets to determine whether they form part of an existing logical connection/file transfer or a new one by searching its database records.   New records are created for new logical connections/file transfers whereas existing records are updated for existing logical connections/file transfers.
5   Thus, the network monitoring computer builds up a historical database of information relating to logical connections and file transfers in the network.

Also of interest is the paper entitled "Usage-Based Cost Recovery in Internetworks" by Ruth and Mills, published in Business Communications Review, Vol. 22, No.7, 1st July 1992, pages 38-42. This paper describes various
10   approaches to charging for access to the Internet. The document describes why flat-rate accounting mechanisms are so prevalent and why usage-based accounting is so complex. The document describes a number of approaches to usage-based accounting all of which involve packet monitoring or analysing.

The document assumes that the Internet is basically a "connectionless"
15   environment. Interestingly, on page 39, left hand column, at paragraphs 2-3 the author mentions the existence of an identifiable connections used by protocols such as TCP/IP, but discounts the possibility of usage-based accounting using these connections because "the higher-level protocol headers are not processed by the network equipment where accounting takes place".

20   Summary of the Invention

Embodiments of the present invention are set out in the appended claims and relate to a method of and apparatus for generating data for charging a user for access over a communications network link by monitoring changes in the state of logical connections created between a client computer system and a server
25   computer. In particular, the logical connections monitored are logical connections

created between a client computer system and a server computer system at the network and transport layers of the International Standards Organisation data communications referenced model (ISO reference model). These logical connections are uniquely defined by four addresses in the ISO reference model.

5   These addresses are: a client network layer address, a client transport layer address, a server network layer address and a server transport layer address. In accordance with the invention, data is recorded in response to the changes of state of the connections detected by the monitoring and charging data is then generated on the basis of the recorded data.

10   In a two party scenario, where a user connects directly to a content provider, the user's computer system acts as the client and the content provider's computer system acts as the server.

In a three party scenario, where a user connects to an access provider which in turn connects to a content provider, there are two client/server

15   relationships. Firstly, between the user's computer system acting as client and the access provider's computer system acting as server and, secondly, between the access provider's computer acting as client and the content provider's computer acting as server. In this scenario, the access provider's computer system is referred to as a proxy server and/or a proxy client.

20   In a further scenario, a user connects directly to a provider of both content and access to further content providers. This content/access provider may in turn connect to a further content provider depending on the user's request. In this scenario, there are either one or two client/server relationships depending on the user's request.

The logical connections are established between a client and a server to provide a logical conduit through which information may be passed. The client typically sends a request to the server which carries out the request and responds to the client. Logical connections are normally established specifically for a single

5   request-response transaction or for a plurality of related request-response transactions and terminated after it or they have finished. During the lifetime of a logical connection, any number of data packets may be sent and received through it depending on the nature of the request-response transaction or transactions.

By monitoring changes in the state of logical connections, such as the

10 establishment and termination of the logical connections between a client and a server, data may be recorded such as the number of logical connections established and terminated, the time and date such connections are established and terminated, the duration of such connections, and any of the four addresses uniquely identifying the logical connection i.e. the client network layer address, the

15 client transport layer address, the server network layer address and the server transport layer address. Additional data may be recorded, if required, such as the number and size of data packets sent and received over such logical connections and information extracted from the header information prepended to the data packets. Charging data may then be generated on the basis of the recorded data.

20   The present invention offers various advantages over different ones of the solutions described in the prior art. One problem for content providers and access providers, particularly for those operating in the field of the Internet, is determining how to charge a user appropriately for the use he makes of a connection over a "dial-up" communications link. Another problem for content providers and access

providers is determining how to charge a user appropriately for the use he makes of a connection over an "always-on" communications link.

Another solution suggested in the prior art, whether for connection via an "always-on" or a "dial-up" communications links, is for the user to dial up a
5   separate PSTN telephone call to enable the user to access content, to provide a meaningful measurement of "on-line" time or to allow appropriate charges to be set by the access or content provider through a special telephone number charging mechanism.

The present invention provides a flexible mechanism for generating
10 charging data depending on the use made of connections over the communications network link, without the need for the complex, processor-intensive task of measuring data packet throughput, and without the need for a "dial-up" PSTN telephone call. The present invention may be applied to "dial-up" communications links as well as "always-on" communications links. However, preferable
15 embodiments of the invention involve "always-on" communications links where the problem of appropriately charging for usage is more acute.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Description of Drawings

20   Figure 1 is a diagram showing a prior art communications network system;

Figure 2 is a diagram showing an embodiment of a communications network system in accordance with an embodiment of the present invention;

Figure 3 is a diagram showing logical connections established between a number of client computer systems and a multi-homed server computer system;

Figure 4 is a flow diagram showing user / access provider / content provider interactions in accordance with the present invention;

Figure 5 is a diagram showing the timeline of a number of logical connections in a client/server interaction; and

5   Figure 6 is a diagram showing a typical record created in response to monitoring logical connections in accordance with the present invention.

Description of Typical Client/Server Logical Connections

Application Programming Interfaces (APIs) for programming clients and servers to handle logical connections are well-established and well-documented.

In one embodiment of the present invention, in the field of the Internet, 5 the logical connections are Transmission Control Protocol/Internet Protocol (TCP/IP) socket connections and the API are the Sockets API originally developed by the Computer Systems Research Group at the University of California at Berkeley. The Sockets API provides well-documented routines for handling TCP/IP socket connections. For a detailed description of TCP/IP and the Sockets API see W. 10 Richard Stevens' 'TCP/IP Illustrated Vols. 1-3'.

Each socket connection is uniquely defined by four addresses: the client IP address (i.e. the client network layer address), the client port number (i.e. the client transport layer address), the server IP address (i.e. the server network layer address), and the server port number (i.e. the server transport layer address).

15 IP addresses uniquely identify interfaces on particular computer systems connected to the Internet. For example, if the user were to connect directly to a content provider using a browser such as Netscape Navigator (TM) or Microsoft Internet Explorer (TM), the client network address would be the IP address of the user's computer system and the server network address would be the IP address 20 of the content provider's computer system (or one of the IP addresses of the content provider's system where the content provider operates a multi-homed server).

Transport layer addresses are used in addition to IP addresses to identify particular connections. If the user directly connected to the content provider above 25 were to access hypertext documents from the Web, the server transport layer

address would be TCP port number 80 which is the well-known port for the Hypertext Transfer Protocol (HTTP). The user transport layer address would be an "ephemeral port number" i.e. a unique but arbitrary number from 1024 to 5000 assigned by the user's computer system to uniquely identify one client connection

5 request.   With some browser clients, several ephemeral ports are used simultaneously to speed up access of complex content such as multi-file Web pages.

The TCP/IP is a protocol suite that allows different computer systems to communicate with each other over the Internet. The various protocols that make

10 up TCP/IP are arranged in a layered structure. TCP/IP is normally regarded as a four layer protocol suite with a link layer at the bottom, a network layer above that, a transport layer above that and an application layer at the top. The link layer normally includes the physical communications hardware plus device drivers to operate it. The network layer handles the movement of data packets around the

15 network.   IP, the Internet Protocol, is the most important protocol at this layer. Each network interface on each computer system connected to the Internet has a unique IP address. This enables IP to send data to the correct destination even though the data may take many different routes to get there. The transport layer handles the data to be sent over the Internet. It is responsible for dividing the data

20 into appropriately sized packets for transmission. The two dominant protocols at this layer are TCP and UDP. TCP, the Transmission Control Protocol, provides a reliable flow of data by requiring the receiving computer system to acknowledge receipt of data packets sent.   However, UDP, the User Datagram Protocol, provides a simpler service in which data packets, known as datagrams, are

transmitted with no guarantee that they will arrive at their destination. The application layer handles the particular service being used.

There are many different TCP/IP applications available, for example, the File Transfer Protocol (FTP), the Simple Mail Transfer Protocol (SMTP), the Simple

5   Network Management Protocol (SNMP), the Trivial File Transfer Protocol (TFTP), the Hypertext Transfer Protocol (HTTP), and the Network News Transfer Protocol (NNTP). Different applications use different transport layer protocols. FTP, SMTP, HTTP and NNTP use TCP, for example, whereas TFTP and SNMP use UDP. TCP and UDP identify which applications data packets are intended for using 16 bit port

10   numbers.   Applications are normally associated with well-known port numbers. For example, every TCP/IP server offering FTP provides service on TCP port 21, and every server offering TFTP provides service on UDP port 69.

When an application sends data over the Internet, the data is sent down the protocol stack. Each layer adds required control information by prepending

15   headers to the data it receives. For example, IP adds header information including the source and destination IP addresses and both TCP and UDP add information including the source and destination port numbers. The process of prepending successive headers to the data is called encapsulation. The process of reading and removing the headers at the receiving end is known as demultiplexing.

20   Description of Prior Art Charging System

Figure 1 shows an example, according to the prior art, of how a user's computer system 10 may be connected to a number of content provider's computer systems 20 over the Internet 40 via an access provider's computer system 30. The user's computer system 10 is connected to the access provider's

25   computer system 30 by means of, for example, a pair of modems 50 transmitting

and receiving signals over copper twisted-pair telephone lines 60. In some instances of the prior art, the user also makes an ordinary PSTN telephone call from a telephone 70 to the access provider 30 via a telecommunications operator 80. Charging information may be generated by the access provider 30 and stored

5   in a data storage means 90.

Detailed Description of the Invention

Figure 2 shows how a user's computer system 10 is connected to a content provider's computer systems 20 via an access provider's computer system 30 in one embodiment of the present invention according to the three-party

10   scenario described above.

The access provider's computer system 30 is connected to the content provider's computer system 20 through the Internet 40, for example, or some other general communications network. The user's computer system 10 is connected to the access provider's computer system 30 by means of a pair of

15   xDSL terminal units 50 transmitting and receiving digital signals over a twisted-pair telephone line 60, for example. The remote xDSL terminal unit is connected to the access provider's computer system 30 through a permanent virtual connection in an Asynchronous Transfer Mode (ATM) network 70, for example. The permanent connections between the user's computer system 10 and the access provider's

20   computer system 30 over the ATM network 70 are managed by the network management system 80.

In a further embodiment of the present invention, according to the two-party scenario described above, computer system 30 is a content provider and is able to provide content to the user's computer system 10 via the permanent virtual

25   connection over the ATM network 70, for example.

In a further embodiment of the present invention, according to the further

scenario described above, computer system 30 is a provider of both content and

access. Computer system 30 is connected and able to provide content to the

user's computer system 10 as described above in both the embodiments according

5   to the two- and three-party scenarios.

In all three embodiments, computer system 30 monitors logical

connections made between the user's computer system 10 and itself and may

monitor logical connections made between itself and the content provider's

computer systems 20, if required. Data is created in response to changes in the

10 states of these logical connections detected by the monitoring and is recorded in a

data storage means 90. Charging data generated on the basis of the recorded data

is also stored in the data storage means 90, or may be recorded in a further data

storage means (not shown).

Figure 3 shows a number of logical connections that may be created

15 between a multi-homed server 110 and a number of clients 120, 122, 124 and

126 in the field of the Internet. With a multi-homed server design, server 110 may

have two IP addresses 112 and 114 at the same computer system. Also, server

110 may offer a number of different services operating out of different ports 150

and 152 each having different port numbers. Also, since server 110 is a multi-

20 homed server, it may offer one of the same services out of port 154 as out of

ports 150 or 152, or it may offer a different service with a different port number.

Furthermore, server 110 may offer the same service operating out of the same

port, for example port 150, to a number of clients 120, 122 and 124.

Servers may be concurrent or iterative. Iterative servers only handle the

25 requests of one client at a time, whereas concurrent servers can handle multiple

requests from multiple clients simultaneously. In the Sockets API implementation, a typical concurrent server will create a passive socket 130 and bind it to the port 150 of a well-known service. Socket 130 will be placed in a listening state and will be able to accept requests to establish socket connections with another socket

5   140 created by client 120 and bound to a port 160 for example. Port 160 is not associated with a particular service or request, but is an "ephemeral" port assigned a unique port number on that client.

Socket connections 180 and 182 show how two clients 122 and 124 with ports 162 and 164 may be connected to server 110 operating a well-known

10   service out of port 150 via two sockets 132 and 134 at the server, both bound to port 150, and via two sockets 142 and 144 at the two clients. The server sockets 132 and 134 are known as active sockets because they are actively connected to client sockets. In a concurrent server, these two server sockets are normally handled under separate processes or threads. When a passive socket 130 bound

15   to a port 150 of a well-known service receives a request to establish connections with a client socket 140, the server normally spawns a new process or thread with a new socket to provide the service leaving itself free to accept further requests from further clients. Once the client request has been dealt with, the active socket between the client and the server is dropped. This system allows a concurrent

20   server to service more than one client at the same time and on the same port. Concurrent servers may also offer more than one service at the same time by creating a number of passive sockets bound to the well-known ports of the services offered. Normally, the individual sockets would be handled under separate processes or threads.

Additionally, a server may create two sockets 136 and 138 bound to the port 154 of a well-known service and connected to two sockets 146 and 148 bound to two ephemeral ports 166 and 168 of one client 126. In this case, the socket connections 184 and 186 are uniquely identified despite the fact that the
5   same client has established connections with the same server on the same server port because the ephemeral client ports 166 and 168 are represented by unique port numbers.

Figure 4 shows an interaction between the computer systems of a user, an access provider and a content provider in a Sockets API implementation. In this
10   illustration, the access provider's computer system and the content provider's computer system are operating a TCP connection-oriented service such as HTTP or NNTP and are designed as iterative servers. The boxes represent the routine calls made in the Sockets API to perform operations on TCP sockets. Boxes appearing under different columns but on the same line represent events occurring at
15   different computer systems but at about the same time.

In the three party scenario, the access provider's computer system, acting as a proxy server, creates a socket by calling the socket() routine call at line 1. At line 2 the access provider's computer system binds the socket to a port. At line 3 the access provider's computer system puts the socket in a listening state and at
20   line 4 the access provider's computer system calls the accept() routine call which blocks the process until a request for connection is received. Up to this point, the socket created by the access provider's computer system is in a passive state, i.e. it has not established a connection with another socket.

Similarly, the content provider's computer system creates a socket, binds it to a port, puts it in a listening state and calls the accept() routine call for its own socket (lines 1 to 4).

Meanwhile, the user's computer system creates a socket at line 5 and 5 calls the connect() routine call at line 6. The Web browser client might bind the socket to a port itself but normally leaves this to the kernel of the user's computer system since the socket must be bound to a unique ephemeral port in any event. The user connect() routine call attempts to establish a socket connection with a server at a particular IP address and on a particular port representing a well-known 10 service that the user is attempting to access.

The user's client program will be set up to connect to the proxy server of the access provider. For example, with Web browser client programs such as Netscape Navigator (TM) or Microsoft Internet Explorer (TM), there are normally options to set up the appropriate addresses for proxy servers within the browsers' 15 configurations.

The user's connect() routine call attempts to establish a connection with the access provider's computer system by sending and receiving data packets which correspond to a three-way handshake. The user's computer system first sends a TCP 'SYN' (synchronise) flag to the access provider's computer system 20 which responds with a SYN flag of its own and an 'ACK' flag to acknowledge the user's SYN. Finally, the user's computer system responds with an ACK flag to acknowledge the access provider's SYN. The SYN flags serve to synchronise data packet sequence numbers maintained at client and server. These sequence numbers allow the TCP/IP stacks to fragment the data to be transmitted into a

number of sequenced data packets of a suitable size and reassemble them again after transmission.

Once the three-way handshake is complete the TCP socket connection is established. Once the user's computer system has established connections with

5   the access provider's computer system, the access provider's accept() routine call returns, i.e. it stops blocking and continues executing the process. At line 7 the user's computer system forwards one or more data packets to the access provider's computer system, for example a request to get a Web page from the content provider.   This is achieved through a write() routine call at the user's

10 computer system and a read() routine call at the access provider's computer system.   At line 8 the access provider's computer system, this time acting as a proxy client, creates a new socket and at line 9 attempts to connect it to the content provider's computer system as described above.   If the connection is successfully established, the accept() routine call of the content provider's

15 computer system returns at this point.   Thus, the access provider's computer system now has two sockets actively connected to sockets at the user's computer system and at the content provider's computer system.

At line 10 the access provider's computer system forwards the user's data packets, eg the Web page request, to the content provider's computer system.   At

20 line 11 the content provider's computer system sends one or more data packets to the access provider's computer system in response, for example the requested Web page.   At line 12 the access provider's computer system forwards the data packets to the user's computer system which, for example, displays the Web page.

At line 13 the user's computer system and the access provider's computer system close their sockets and thus terminate the connection between them, and at line 14 the access provider's computer system and the content provider's computer system close their sockets and terminate the connection between them.

5 TCP normally terminates connections by sending a 'FIN' (finished) flag from each end of the connection and an ACK flag to acknowledge the FIN. Sometimes, TCP terminates connections by sending an 'RST' (reset) flag instead of a FIN flag.

This process is typical where both the access provider and the content provider are operating iterative servers. Where the servers are concurrent, a new

10 process is spawned after the accept() routine call returns in each case. In the new process, a new socket is created and connected to the client socket to provide the service to the client whilst the passive socket remains free to accept requests for socket connections from other clients.

Figure 4 may be adapted to show the typical interactions between the

15 computer systems of a user and a content provider in the two-party scenario described above by removing lines 8 to 11, removing the column entitled 'Content Provider' and renaming the column entitled 'Access Provider' to 'Content Provider'.

Figure 4 may be adapted to show the typical interactions between the computer systems of a user, a content/access provider and a further content

20 provider in the further scenario described above by renaming the column entitled 'Access Provider' to 'Content/Access Provider' and adding parentheses around lines 8 to 11 to represent the potential connection between the content/access provider's computer system and the further content provider's computer system.

In all cases, whether in the two- or three- party scenario or the further scenario and whether the servers are iterative or concurrent, the present invention is implemented by the addition of certain program routines in a server program.

In the three party scenario, routines are inserted in the access provider

5   server program in between the steps of lines 6 and 7 to record data concerning the establishment of the connection with the user's computer system and in between the steps of lines 13 and 14 to record data concerning the termination of the connection with the user's computer system. Similarly, routines are inserted to record data concerning the establishment of a connection with the content

10 provider in between the steps of lines 9 and 10 and the termination of the connection with the content provider after line 14, if required.

In the two party scenario, routines may be similarly inserted at corresponding places in the content provider server program to record data concerning the establishment and termination of connections with the user's

15 computer system.

In the further scenario, routines may be similarly inserted at corresponding places in the content/access provider server program to record data concerning the establishment and termination of connections with the user's computer system and/or the further content provider's computer system.

20 Routines are inserted into server programs to record a variety of data concerning the establishment and termination of socket connections such as the number of socket connections established and terminated with the user's computer system, the time and date of establishment and termination of connections with the user's computer, and the duration of socket connections with

25 the user's computer system. Details of the IP address of the user's computer

system and the IP address and port number, representing the service provided, of the content provider's computer system may also be recorded. Information may also be extracted and recorded from the TCP/IP headers prepended to the data packets passed. Which and how many of such routines are inserted will depend on

5 the data desired to implement a particular charging mechanism.

Figure 5 shows the timeline of a number of logical connections in a typical two party client/server interaction between a user's browser and a content provider's HTTP server over the Internet. When using HTTP version 1.0 browsers such as Netscape Navigator sometimes create a number of connections, usually up

10 to 4 at a time, to download data quickly. Here, for the purpose of illustration, a browser has created 8 connections in total to download a Web homepage and 7 graphic images. The 8 separate bars in Figure 5 represent the 8 connections. At row A, the first connection is established between the browser and the HTTP server to get a basic Web HTTP homepage file, for example. In this example, the

15 HTTP homepage file refers to 7 associated graphic image sources. Once the HTTP homepage file has been sent, the connection is terminated. Immediately thereafter, the browser establishes 4 connections in parallel at rows A, B, C, and D to download the graphic image files from the HTTP server. Once each of these are downloaded, the connections are immediately terminated and new connections

20 established to download the remaining images, if required. The content provider's computer system monitors changes in the state of these connections as described above and record data in response to the monitoring as described in more detail below.

In the three party scenario, the browser establishes connections with the

25 proxy server of an access provider which in turn establishes corresponding

connections with the HTTP server of the content provider. Thus, in this example there would be 16 connections in total. The access provider's computer system monitors changes in the state of these connections as described above and records data in response to the monitoring as described in more detail below. In one

5   embodiment of the present invention, the access provider's computer system monitors and records data concerning only the connections with the user's browser, or, in another embodiment, data concerning only the connections with the content provider's server, or, in a further embodiment, data concerning both sets of connections. The choice of embodiment depends on the charging

10 mechanism preferred.

In the further scenario, the browser establishes connections with the proxy server of an content/access provider which may, depending on the user's request, in turn establish corresponding connections with the HTTP server of a further content provider. The content/access provider's computer system monitors

15 changes in the state of these connections as described above and records data in response to the monitoring as described in more detail below. In one embodiment of the present invention, the content/access provider's computer system monitors and records data concerning only the connections with the user's browser, or, in another embodiment, data concerning both the connections with the user's

20 computer system and the further content provider's server. The choice of embodiment depends on the charging mechanism preferred.

HTTP 1.1, a later version of HTTP 1.0, differs from HTTP 1.0 in a number of respects. One respect of relevance to the present invention is the inclusion of persistent connections and pipelining.

HTTP 1.1 defines persistent connections as to the default method for connections in client/server interactions. Instead of opening and closing a single connection for each request-response transaction between a client and a server, an HTTP 1.1 client or server will maintain an opened connection for multiple request-

5   response transactions, until such time as either the client or the server sends a "close" token in the HTTP connection header. This then indicates that the request-response transaction is to be the last for that connection and that once the transaction is completed the connection between the client and the server may be dropped. Both client and server may specify time-outs when using persistent

10 connections in HTTP 1.1 so that idle connections are not maintained indefinitely.

HTTP 1.1 persistent connections also allow pipelining of requests and responses to requests so that a client may send multiple requests to a server without the need to wait for a response to each request to arrive before issuing a new request.

15   HTTP 1.1 still allows clients and servers to establish multiple parallel connections, but the advantage of doing this is not so great as with HTTP 1.0 because the combination of persistent connections and pipelining requests and responses already improves the performance of the connections. The number of parallel persistent connections established between a client and a server in HTTP

20 1.1 is normally limited to a maximum of two.

In the three party scenario, where a client connects to a proxy server of an access provider using HTTP 1.1 persistent connections, and the proxy server in turn connects to the server of a content provider, then the proxy server of the access provider should also use persistent connections in its connection with the

25 content provider's server.

In the further scenario, where a client connects to a proxy server of an content/access provider using HTTP 1.1 persistent connections, and the proxy server in turn connects to the server of a further content provider, then the proxy server of the content/access provider should also use persistent connections in its

5   connection with the further content provider's server.

Monitoring the changes in the state of HTTP 1.1 persistent connections, in the two or three party scenario and in the further scenario, is carried out in the same way as monitoring changes in the state of HTTP 1.0 connections described above.

10   Figure 6 shows a typical data record 220 that may be created in respect of a single logical connection. Data storage means 210 may be used to store such records. For example, in the field of the Internet, the User/Content Provider Identifier item represents the IP address of the user's computer system and, in the three party scenario, the IP address of the content provider's computer system.

15   The Service Identifier item represents the port number of the well-known service offered by the access or content provider. The Start Time/Date item represents the point at which the logical connection was established and the End Time/Date item represents the point at which the logical connection was terminated. The Duration item represents the period of time during which the logical connection

20   was established. The Extracted Information item represents further information that may be extracted from the data packets sent and received over the logical connection. For example, this item might include details of URL (Uniform Resource Locator) of the page requested and received, the size of those data packets, and other information extracted from the headers prepended to the data packets as

25   required.

In other embodiments of the present invention various combinations of the recordable items listed above are recorded.

In an embodiment recording only the User Identifier, the charging data generated would simply depend on the number of logical connections created.

5   In other embodiments in which other combinations of the recordable items are recorded, the charging data generated may depend on other factors such as the duration of logical connections, the start time and date or end time and date of any such logical connections, and the type of service offered to the user. For example, in the field Internet, a content provider or an access provider might want

10 to charge differently for the HTTP service than for the NNTP service.

In one embodiment at least the User Identifier, the Service Identifier, the Start Time/Date, and the End Time/Date is recorded. From this data, the duration of each logical connection can be deduced. With this data, the charges could be generated based on the total duration of the logical connections and the type of

15 service provided to the user, with different rates possibly being applied to different periods of the day or other periodic cycles.

Additionally, if more charging data is required, more detailed information may be recorded such the number and size of data packets sent and received, and the information extracted from headers prepended to those data packets. For

20 example, information may be extracted from the IP header, UDP header, TCP header, HTTP header and NNTP header.

In a further embodiment of the present invention, the program routines inserted into the server program do not record data for the logical connections individually, but record data relating to the cumulative duration of those

25 connections within a predefined period of time.

In this embodiment, the routines inserted into the server program between the steps in Figure 4 of lines 6 and 7, 9 and 10, 13 and 14 and after line 14 described above start and stop a number of timers maintained by the computer system when logical connections are established and terminated respectively. In

5   the case where more than one logical connection is established between the user's computer system and the access or content provider's computer system within the predefined time period, for example within a 24 hour period, a corresponding number of timers need to be maintained by the access or content provider's computer system. For example, a number of different users' computer systems

10   may establish logical connections with the access or content provider's computer system and, furthermore, each user's computer system may establish more than one logical connection with the access or content provider's computer system in parallel.

At the end of the predefined time period, the access or content provider's

15   computer system would create a record of the cumulative duration of all logical connections established with a user's computer system for each individual user's computer system. Thus, a record would contain two items of information: the Client Identifier, and a Cumulative Duration item. At the end of each predefined time period, the access or content provider's computer system also resets all

20   timers currently recording time so that the records created for the next predefined time period do not include time recorded for the time period ending.

One advantage of the above embodiment is that only one record is created for each user's computer system which establishes logical connections with the access or content provider's computer system within the predefined time period.

25   This is likely to be a relatively lower number of records than in embodiments where

a record is created for each logical connection established between each user's computer system and the access or content provider's computer system as described above.

Finally, charging data would be generated on the basis of the recorded

5   data by means of a rating engine processing the recorded data and bills or invoices could then be generated by means of a billing or invoicing engine.

It will be apparent that there are alternative implementations of the invention in which data is recorded in response to monitoring changes in the state of logical connections at the network and transport layer of the OSI reference

10 model between a client and a server and in which charging data is then generated on the basis of the recorded data.

The Sockets API is the standard API for developing client and server network communications programs for the Internet. However, the present invention has application to other data communication networks in which an

15 access provider or a content provider wishes to charge a user for access. Accordingly, other embodiments of the present invention would involve monitoring logical connections using TCP/IP or other protocols and handled using other network communications APIs including the Transport Layer Interface (TLI) API, the Common Programming Interface for Communications (CPI-C) API, and the

20 Named Pipes API, for example.

Communication using protocols other than TCP/IP could also be monitored and charged for using the present invention. Such other protocols include the Internet Packet Exchange/Sequenced Packet Exchange (IPX/SPX) protocol and the Wireless Application Protocol (WAP), for example.

It can be seen from the above that the present invention provides a flexible solution to the problems of appropriately charging a user of a computer system for access to content provided over a communications network either directly or via an access provider. The solution avoids the disadvantages of the prior art and provides a mechanism for generating charging data which is flexible, appropriate to the level of usage, easy to implement, low cost in terms of processor operations, and invisible to the user.

CLAIMS:

1.   A method of generating charging data relating to information passed between a user's computer system and a content provider's computer system over

5   a communications network link involving at least one client/server relationship, the method comprising the following steps:

a)   monitoring communications in at least one client/server relationship to detect at least one event causing a change in a state of at least one logical connection defined by at least a client network layer address, a client transport layer address,

10   a server network layer address and a server transport layer address,

b)   recording data created in response to the at least one event detected by the monitoring, and

c)   generating charging data on the basis of the recorded data.

15 2.   A method according to claim 1 wherein the information is passed via an access provider's computer system acting as a proxy client and a proxy server and wherein the monitoring is performed by the access provider's computer system.

3.   A method according to claim 2 wherein the charging data is generated by

20   the access provider's computer system.

4.   A method according to claim 3 wherein the at least one logical connection comprise a logical connection created between the access provider's computer system acting as a proxy server and the user's computer system acting as a client.

5.   A method according to claim 3 or 4 wherein the at least one logical connection comprise a logical connection created between the access provider's computer system acting as proxy client and the content provider's computer system acting as a server.

5

6.   A method according to claim 1 wherein the monitoring is performed by the content provider's computer system.

7.   A method according to claim 6 wherein the charging data is generated by

10   the content provider's computer system.

8.   A method according to claim 7 wherein the at least one logical connection comprise a logical connection created between the content provider's computer system acting as a server and the user's computer system acting as a client.

15

9.   A method according to any preceding claim wherein the recorded data comprises a record of at least one selected from the group consisting of the client network layer address, the client transport layer address, the server network layer address and the server transport layer address.

20

10.   A method according to any preceding claim wherein the recorded data comprises a record of a period of time relating to the at least one event.

11.   A method according to any preceding claim wherein the at least one event comprise the establishment and/or termination of the at least one logical connection.

5 12.   A method according to any preceding claim wherein the recorded data comprises a record of a number of logical connections established and terminated.

13.   A method according to any preceding claim wherein the at least one logical connection comprise one selected from the group consisting of
10 Transmission Control Protocol (TCP) socket connections, User Datagram Protocol (UDP) socket connections and Internet Protocol (IP) socket connections.

14.   A method according to any preceding claim wherein the recorded data further comprises a record of information extracted from at least one header
15 prepended to information passing between the content provider's computer system and the user's computer system during the subsistence of the logical connections.

15.   A method according to claim 14 wherein the at least one header comprise at least one selected from the group consisting of all Transmission Control Protocol
20 / Internet Protocol (TCP/IP) network layer headers, all TCP/IP transport layer headers and all TCP/IP application layer headers.

16.   A method according to any preceding claim wherein the network layer addresses comprise an IP address and the transport layer addresses comprise a
25 port number.

17.   A method according to any preceding claim wherein the communications network link comprises a permanently activated communications network link.

5   18.   Apparatus for generating charging data relating to information passed between a user's computer system and a content provider's computer system over a communications network link involving at least one client/server relationship, the apparatus comprising the following:

a)   computational means for monitoring communications in at least one
10   client/server relationship to detect at least one event causing a change in a state of at least one logical connection defined by at least a client network layer address, a client transport layer address, a server network layer address and a server transport layer address,

b)   storage means for recording data created in response to the at least one
15   event detected by the monitoring, and

c)   computational means for generating charging data on the basis of the recorded data.

19.   Apparatus according to claim 18 comprising a computer system at an
20   access provider acting as a proxy server and a proxy client whereby the information is passed between the content provider's computer system and the user's computer system, the monitoring being performed by the access provider's computer system.

20. Apparatus according to claim 18 adapted to perform the methods of any of claims 2 to 17.

21. A method of generating charging data relating to information passed between a user's computer system and a content provider's computer system over a communications network link using a TCP/IP protocol and involving at least one client/server relationship, the method comprising the following steps:

a) monitoring communications in at least one client/server relationship to detect at least one event causing a change in a state of at least one TCP/IP socket connection,

b) recording data created in response to the at least one event detected by the monitoring, and

c) generating charging data on the basis of the recorded data.

22. Apparatus for generating charging data relating to information passed between a user's computer system and a content provider's computer system over a communications network link using a TCP/IP protocol and involving at least one client/server relationship comprising:

a) computational means for monitoring communications in at least one client/server relationship to detect at least one event causing a change in a state of at least one TCP/IP socket connection,

b) storage means for recording data created in response to the at least one event detected by the monitoring, and

c) computational means for generating charging data on the basis of the recorded data.

23.   Apparatus according to claim 22 comprising a computer system at an access provider acting as a proxy server and a proxy client whereby the information is passed between the content provider's computer system and the

5   user's computer system, the monitoring being performed by the access provider's computer system.

24.   A method of generating charging data relating to information passed between a user's computer system and a content provider's computer system over

10   a communications network link involving at least one client/server relationship, the method comprising the following steps:

a)   measuring at least one duration relating to at least one logical connection defined by at least a client network layer address, a client transport layer address, a server network layer address and a server transport layer address, and

15   b)   generating charging data on the basis of the at least one duration.

25.   A method according to claim 24 wherein the measuring comprises measuring a plurality of durations each relating to a respective one of a plurality of the logical connections, the logical connections being at least partially

20   contemporaneous and relating to the same client/server relationship.

26.   A method as claimed in claim 24 or claim 25, wherein a server of the client/server relationship incorporates a server program which incorporates routines to perform the monitoring.

25

27.   A method of generating charging data relating to information passed between a user's computer system and a content provider's computer system over a communications network link involving at least one client/server relationship, the method comprising the following steps:

5 a)   monitoring communications in at least one client/server relationship to detect at least one event causing a change in a state of at least one logical connection defined by at least a client network layer address, a client transport layer address, a server network layer address and a server transport layer address,

b)   recording data created in response to the at least one event detected by 10 the monitoring, and

c)   generating charging data on the basis of the recorded data, wherein said monitoring is performed by a computer system forming an end point of the at least one logical connection.
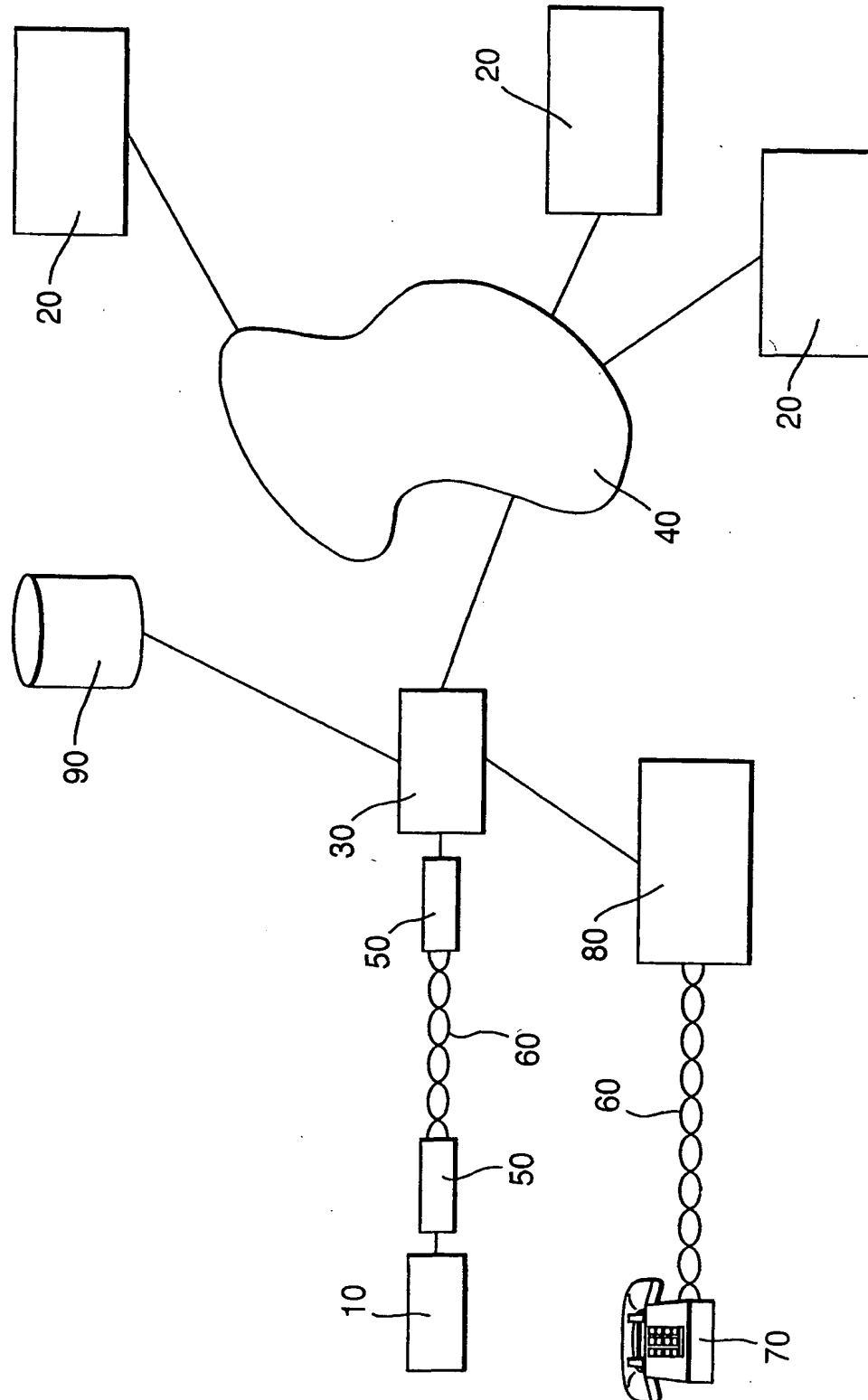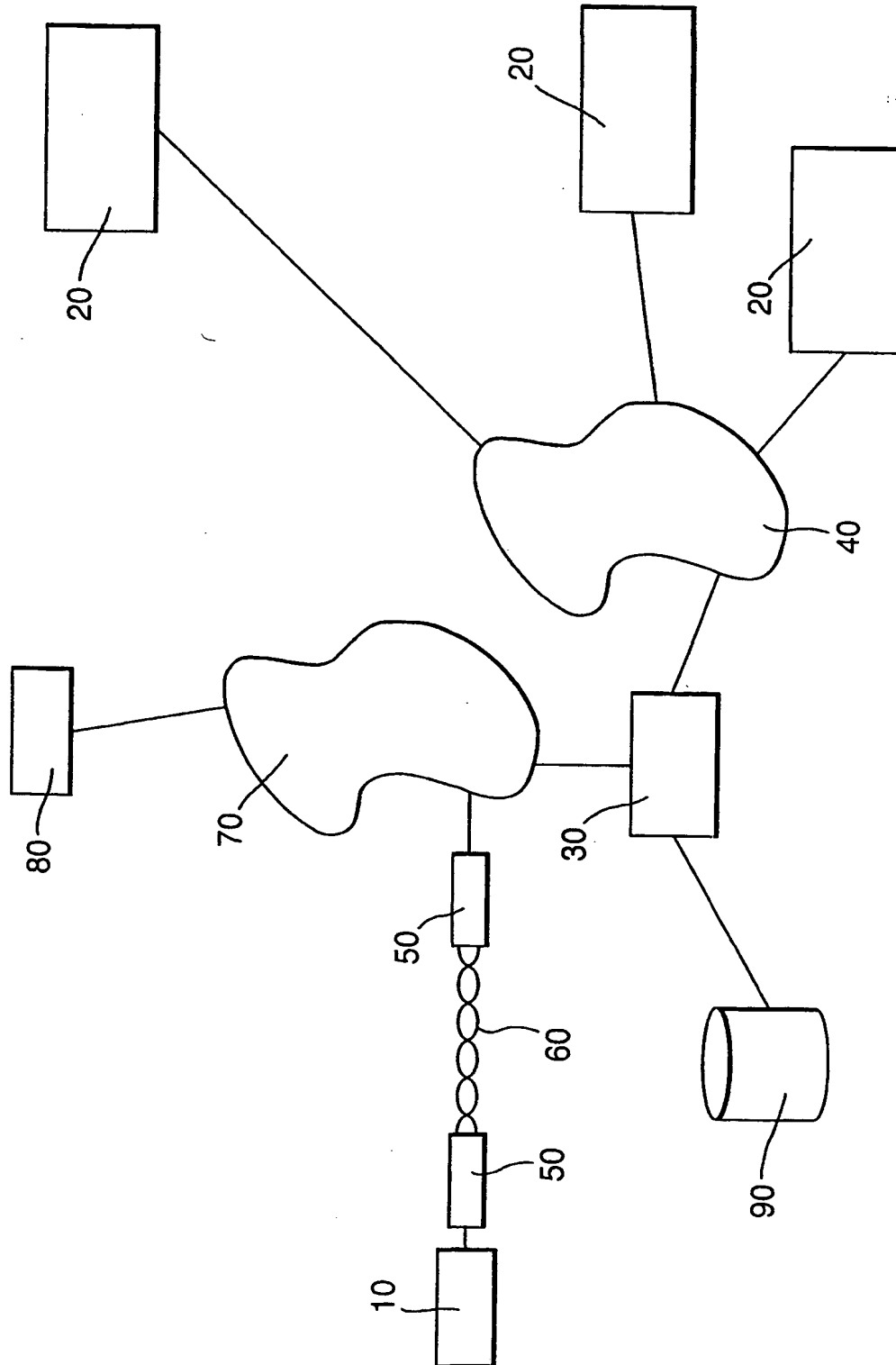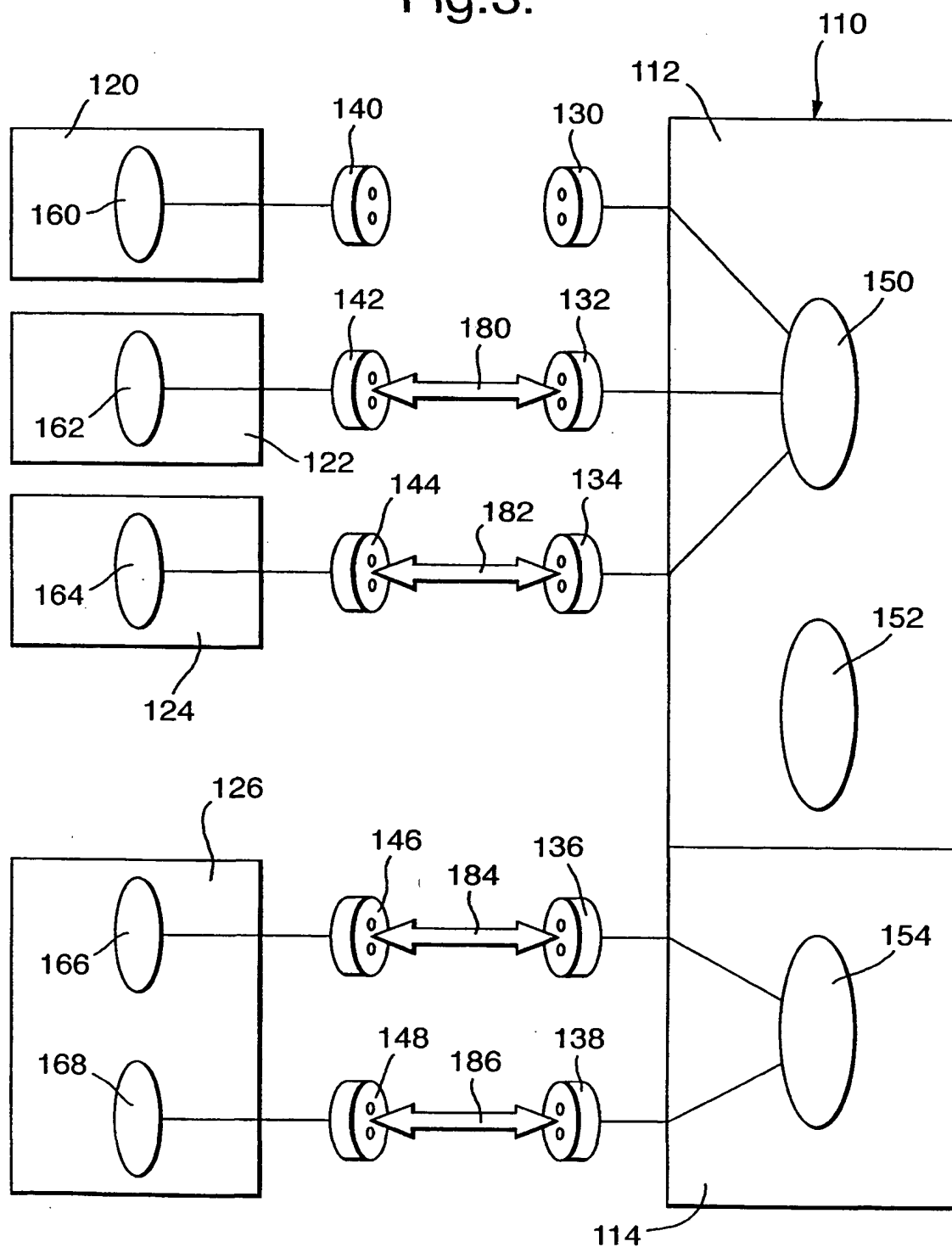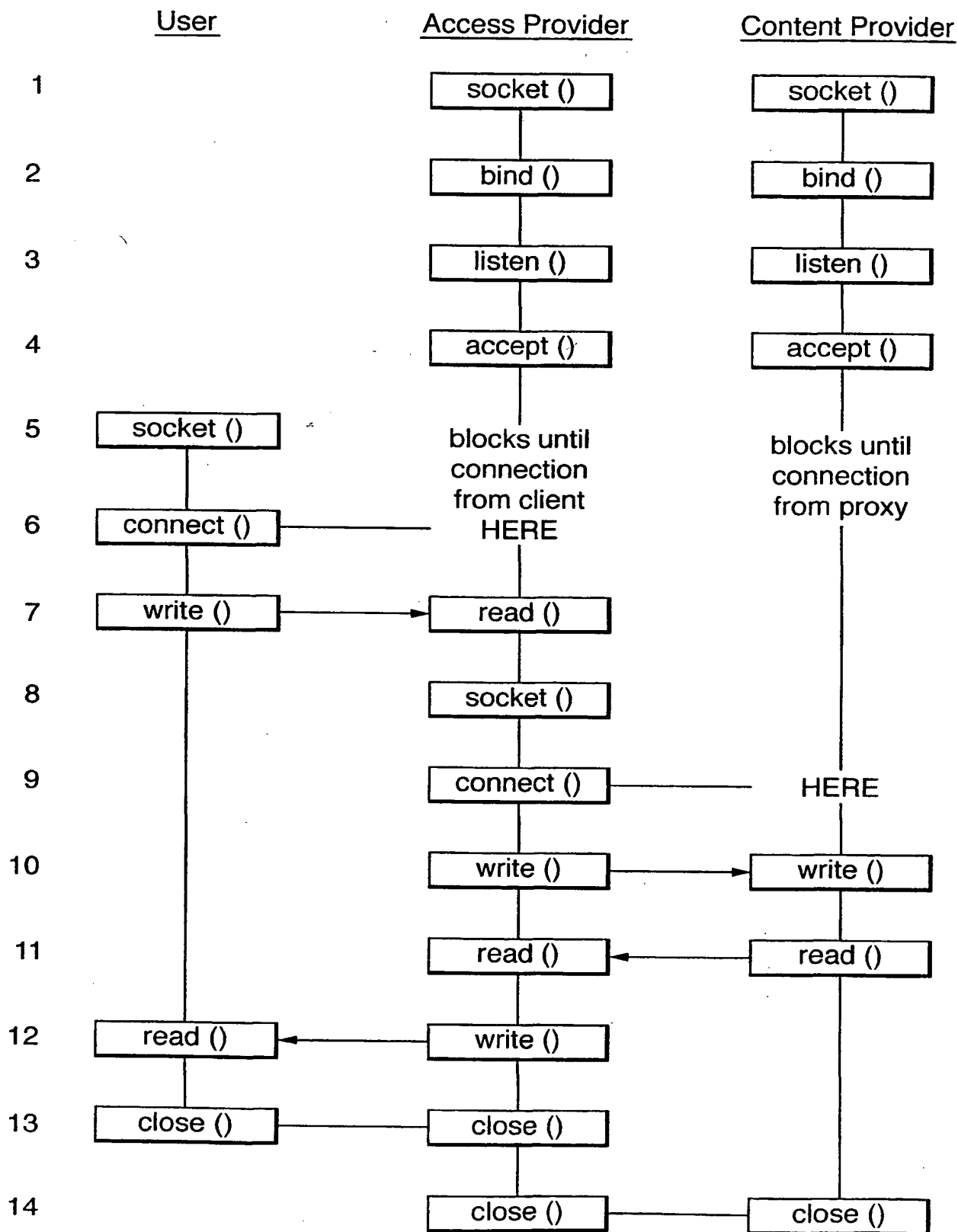
15

Fig.1.

Fig.2.

Fig.3.

4/5

Fig.4.

	User	Access Provider	Content Provider

1   socket ()   socket ()

2   bind ()   bind ()

3   listen ()   listen ()

4   accept ()   accept ()

5   socket ()   blocks until   blocks until
                       connection   connection

6   connect ()   from client   from proxy
                     HERE

7   write ()   read ()

8   socket ()

9   connect ()   HERE

10   write ()   write ()

11   read ()   read ()

12   read ()   write ()

13   close ()   close ()

14   close ()   close ()

Fig.5.

Fig.6.
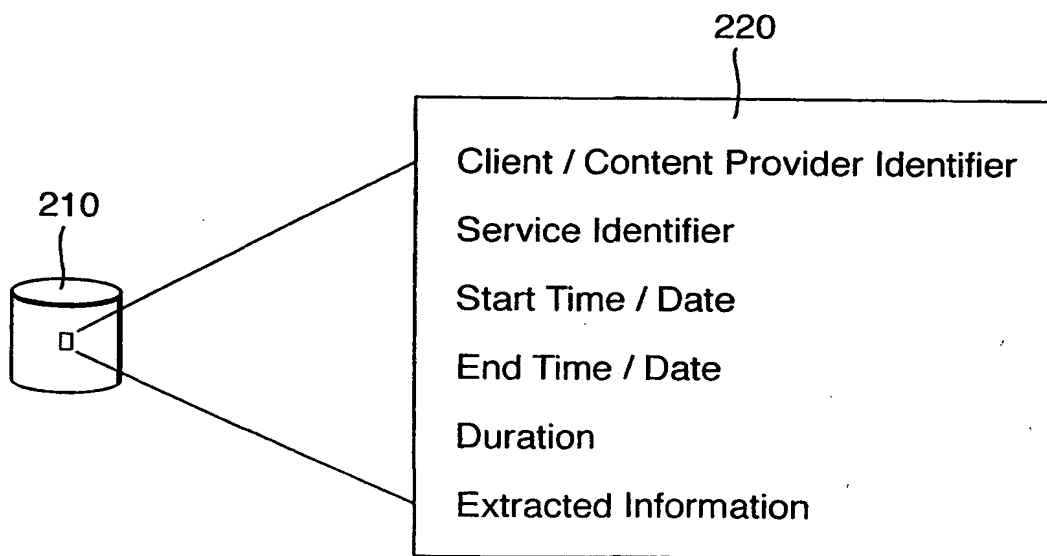
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   H04L12/14   H04L29/06   H04M15/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   H04L   H04M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	US 5 862 335 A (STERNBERG JAY E ET AL) 19 January 1999 (1999-01-19) abstract figures 1-3 column 1, line 47 -column 2, line 10 column 2, line 34 - line 52 column 3, line 26 - line 32 column 3, line 61 -column 4, line 10 column 5, line 21 - line 23 --- -/-	1,9-16, 18,21, 22,24,25

[X] Further documents are listed in the continuation of box C.   [X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
2 June 2000	13/06/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl. Fax: (+31-70) 340-3016	Adkhis, F

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	RUTH G ET AL: "USAGE-BASED COST RECOVERY IN INTERNETWORKS" BUSINESS COMMUNICATIONS REVIEW, vol. 22, no. 7, 1 July 1992 (1992-07-01), pages 38-42, XP000577637 page 38, left-hand column, line 1 - line 8 page 41, left-hand column, line 5 - line 20 page 39, right-hand column, line 18 - line 31	1-25

INTERNATIONAL SEARCH REPORT

Information on patent family members

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5862335	A	19-01-1999	NONE	